

Следует отметить, что использование специализированных программных средств сталкивается с проблемами инсталляции, настройки и обновления систем у конечных пользователей, сложностью передачи данных между уровнями. С другой стороны, в рамках информатизации системы образования практически все школы имеют выход в Интернет и возможности сети практически не задействованы.

В Департаменте образования и науки Тюменской области разработана и внедрена система мониторинга эффективности деятельности образовательных учреждений на базе сети Интернет. Доступ к данным может осуществляться из любой точки, где имеется связь с Интернет, посредством персонального компьютера или мобильного устройства (коммуникаторы, КПК).

Основу системы мониторинга составляет WEB-приложение, устанавливаемое на сервере Департамента образования. Важно отметить, что при этом отпадает необходимость в инсталляции специализированного программного обеспечения на каждом рабочем месте - используется Интернет-браузер, входящий в состав операционной системы. Это также позволяет централизованно обновлять систему мониторинга и наращивать функциональные возможности независимо от конечных пользователей. Особенно это актуально в условиях часто меняющихся показателей деятельности со стороны органов управления образованием.

Разработанный программный комплекс позволяет осуществлять многоуровневый доступ к показателям эффективности деятельности образовательных учреждений. В системе задано несколько уровней доступа: уровень департамента образования, уровень муниципального органа управления образованием и уровень образовательного учреждения. Доступ к данным разрешен только авторизованным пользователям, для которых определены права на ввод, редактирование или просмотр «своих» или «чужих» данных.

С целью обеспечения целостности уже согласованных показателей реализовано автоматическое задание текущего отчетного периода на уровне образовательного учреждения. Период ввода автоматически задается системой и запрещает изменение показателей после контрольного срока.

Для формирования «бумажных» отчетов предусмотрена возможность экспорта выбранных показателей в формат MS Excel 2003.

Интернет система обеспечивает легкую организацию оперативного доступа к данным на ведомственном и межведомственном уровне, для этого достаточно лишь ввести новых пользователей и назначить права доступа.

Следует отметить что, создавая централизованную систему мониторинга деятельности образовательных учреждений на базе сети Интернет необходимо особо ставить вопросы обеспечения работоспособности программного комплекса, безопасности и целостности данных. Требуется создание специальных служб с функциями защиты и восстановления испорченных данных.

Литература

1. Бакланов А.В. Информационная система мониторинга показателей эффективности деятельности образовательных учреждений. XVI Международная конференция-выставка "Информационные технологии в образовании" ("ИТО-2006").

Прыгунков М.О. КОНТЕЙНЕРНАЯ АРХИТЕКТУРА ГИБКИХ ПРОГРАММНЫХ СИСТЕМ

max-prygunkov@yandex.ru

*Волгоградский государственный университет, кафедра «Системы автоматизированного проектирования и поискового конструирования»
г. Волгоград*

Современные программные системы становятся сложнее, претендуя на решение глобальных задач, например таких, как создание единой системы управления предприятием. При этом автоматизация отдельных операций или отделов фактически исчерпала свой потенциал, а возможность безболезненного объединения нескольких подсистем в единое целое, как правило, требует значительных усилий. Реальная проблема заключается в том, что связи между подсистемами должны быть гибкими, изменяемыми и настраиваемыми. Данное положение усугубляется требованиями бизнеса, который нуждается в высоком динамизме. В результате те связи, которые работали десятилетиями, могут быть мгновенно разрушены, например, какое-то подразделение может отказаться от использования услуг централизованной бухгалтерии, посчитав, что это достаточно неэффективно и дорого.

Одно из основных положений BPR (business process re-engineering) говорит о необходимости переноса точки принятия тактических решений как можно ближе к исполнителям. То есть каждое структурное подразделение (СП) должно быть в максимальной степени самостоятельно, самодостаточно и компетентно в принятии решений и представлять собой как бы отдельное предприятие. Централизованное управление предприятиями постепенно сменяется схемами распределённого управления, где значительная часть вопросов, требующих принятия решений, переходят на уровень

структурных подразделений. Управлять же объединением такого большого количества самостоятельных предприятий традиционными методами весьма проблематично. В этих условиях автоматизированные системы должны обеспечить сборку большого числа отдельных самостоятельных подразделений в единое управляемое предприятие. При этом состав и функции подразделений могут постоянно меняться, что требует высокой гибкости программного обеспечения.

Гибкость программного обеспечения определяется набором типовых частей, из которых можно было бы собирать множество программ.

Объектно-ориентированные языки программирования дали четыре новых механизма использования типовых частей:

- механизм классов, порождающих при выполнении любое количество однотипных объектов, например, ряд однотипных кнопок;
- возможность тиражирования объектов от породившей программы во все новые программы;
- динамически линкуемые библиотеки с порождающими объектами классами;
- механизм сборки программ из компонентов – объектов в процессе выполнения программ.

Первый механизм облегчил развитие систем визуального программирования, при работе в которых, значительная часть программы может быть создана путем отбора мышкой стандартных компонентов.

Второй механизм привел к возникновению объектных систем управления базами данных (СУБД), поставляющих программам не только данные, но и код, обрабатывающий эти данные.

На основе третьего механизма возникли СОМ-технологии, что позволило строить гибкие программы, которые обладают свойством возможного развития при изменении условий их эксплуатации.

Четвертый механизм пока не нашел должной поддержки в инструментальных средствах разработки программ, что и предопределило направление данных исследований.

Исходя из того, что контейнер – это некая надстройка над объектно-ориентированным программированием, можно предположить, что контейнеру должны быть присущи неотъемлемые атрибуты объектно-ориентированного программирования (ООП) – наличие у объектов методов и свойств. Так как контейнер – сущность динамически изменяемая, то необходимо в его структуру заложить возможность добавления или удаления новых свойств или методов, а также, замены свойств или методов другими свойствами или методами.

Введём два вида контейнеров: динамический и статический. Контейнер, как и любая сущность, взаимодействует с окружающим миром, т.е. у него есть входы и выходы. У контейнера есть как минимум два интерфейса, с помощью которых он обрабатывает входящий и исходящий информационные потоки.

Но если есть интерфейсы, с помощью которых обрабатываются внешние потоки, то должны быть интерфейсы, обрабатывающие внутренние, вернее, распределяющие поступившую информацию среди объектов, содержащихся в этом контейнере. Градация на статические и динамические контейнеры осуществляется именно по тому, как контейнер распределяет информацию внутри себя.

Если контейнер для «общения» с вложенными объектами общается через единый интерфейс, то такой контейнер будем называть статическим. Следовательно, объекты, вложенные в контейнер, должны обладать таким качеством, как абсолютно одинаковые интерфейсы работы с внешними потоками. Скорее всего, такие объекты будут обладать одинаковой структурой. Если контейнер работает со всеми вложенными объектами не по одной схеме, а используя разные методы, то такой контейнер будем называть динамическим.

Чтобы приведённая классификация контейнеров стала более понятной, стоит оговориться о том, что разделение контейнеров на статические и динамические относится к строению контейнера. Также следует снова указать на одну из особенностей контейнеров: любые контейнеры могут агрегироваться другими контейнерами, т.е. статический контейнер может состоять из динамических и, соответственно, наоборот.

Для большей информативности мы «нагрузим» наши контейнеры дополнительными атрибутами – событиями. События будут возникать в 6 случаях:

1. значение свойства было изменено;
2. была попытка изменить свойство, но произошла системная ошибка;
3. метод был запущен;
4. метод завершил работу;
5. во время работы метода возникла ошибка;
6. б) была попытка запуска метода, но произошла системная ошибка.

Контейнеры по определению могут общаться с другими контейнерами, с которыми находится на одном уровне иерархии, а также на одну ступень выше и ниже по иерархической лестнице. Это общение происходит путём передачи сообщений. Соответственно, сообщение может быть внутренним

относительно контейнера – ситуация, когда контейнер передаёт сообщения между своими компонентами, и внешними – ситуация, когда контейнер передаёт сообщения другим контейнерам или контейнеру, в состав которого входит источник сообщения.

В общем виде графически контейнер можно представить так, как это показано на Рисунке 1.

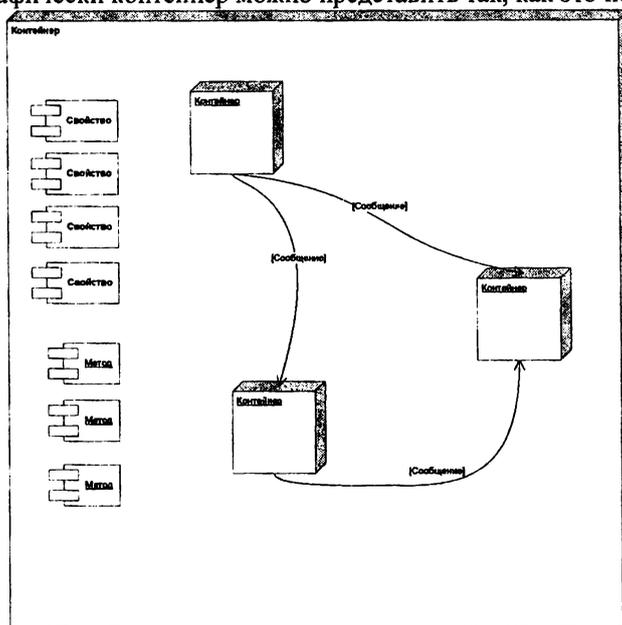


Рисунок 2 – схематическое представление строения контейнера.

Для обеспечения заявленной возможности замены (добавления/удаления) новых свойств и методов принято решение хранить в контейнерах только ссылки на свойства и методы.

Помимо такой программной сущности, как контейнер, в программную систему необходимо внедрить модули-диспетчеры – виртуальные таблицы, в которых содержатся ссылки на экземпляры свойств и методы, обращаясь к которым объект-контейнер фактически агрегирует их в своей структуре.

Итак, контейнер – это некая сущность, которая обладает такими атрибутами, как свойство, метод и событие. Контейнер может состоять из контейнеров и входить в состав других контейнеров. Контейнеры общаются между собой путём передачи сообщений.

Применение контейнеров для построения крупных программных систем, которые должны обладать возможностью быстрого изменения исходя из требований бизнеса, предоставляет увеличивает скорость создания такой системы и существенно облегчает процесс её поддержки.

Пыхтин А.И., Спирин Е.А, Захаров И.С.
ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ СИСТЕМЫ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ В УПРАВЛЕНИЕ ПРИЕМНОЙ КАМПАНИЕЙ ВУЗА

sephiroth_kstu@mail.ru
 Курский государственный технический университет
 г. Курск

Экономическая эффективность внедрения информационных систем в области образования не может быть оценена с точки зрения инвестиционных проектов, так как вуз не является организацией, в которой внедрение информационных технологий напрямую влияет изменение прибыли. Поэтому наиболее подходящей методикой оценки экономического эффекта может стать расчет снижения основных и временных затрат, прежде всего оплаты труда персонала, задействованного в приемной кампании.

Оценим затраты на организацию приемной кампании в условиях отсутствия информационной системы. Исходные данные для расчетов можно заимствовать из опыта работы приемной комиссии КурскГТУ. Динамика подачи заявлений и внесенных изменений в личные дела абитуриентов приведена на рис. **Ошибка! Источник ссылки не найден.** Очевидно, что пиковая нагрузка на сотрудников приемной комиссии приходится на 2-13 недели работы, т.е. в течение 3 месяцев приемной кампании. Именно на этот срок и необходимо набирать персонал. До 2001 г., например, в проведении приемной кампании Курского государственного технического университета (КурскГТУ) было задействовано порядка 50-60 человек профессорско-преподавательского состава кафедр, 10-15 студентов, и работающие круглый год консультанты. Какое-либо специализированное программное и аппаратное обеспечение для обработки информации о потоке абитуриентов отсутствовало, все документы заполнялись вручную.